Education and Information Technologies
https://doi.org/10.1007/s10639-024-13297-4

Redefining computational thinking: A holistic framework and its implications for K-12 education

Belén Palop[1] · Irene Díaz[2] · Luis J. Rodríguez-Muñiz[3] · Juan José Santaengracia[3]

Received: 29 April 2024 / Accepted: 23 December 2024

Abstract

In the realm of K-12 Education, the growing significance of Computational Thinking has sparked extensive inquiry into its nature and instructional methodologies. Despite a wealth of literature on the subject, ongoing debates persist regarding its fundamental components. Across global educational landscapes, Computational Thinking is being incorporated into curricula with varying degrees of emphasis, ranging from robotics to computer coding to broader algorithmic problem-solving approaches. This article aims to reassess established definitions of Computational Thinking and propose a nuanced understanding that highlights the necessity and importance of data in contemporary problem-solving contexts. Additionally, we present a comprehensive framework aimed at cultivating Computational Thinking skills in students, encompassing technical competencies alongside cognitive and metacognitive processes. Subsequently, we offer (1) a refined definition emphasizing the importance of data, (2) a comprehensive framework outlining essential components to foster Computational Thinking skills in students, and (3) a discussion of these concepts within the context of K-12 education.

Keywords Computational thinking · Computer science education · Data · K-12 education · Mathematics education

✉ Juan José Santaengracia
juanjose@uniovi.es

[1] Department of Soc., Exp. Science and Mathematics Education, Fac. Educación, Rector Royo-Villanova, Complutense University of Madrid, Madrid 28040, Spain

[2] Department of Computer Science, University of Oviedo, Campus de Llamaquique, Oviedo 33007, Spain

[3] Department of Statistics & O.R. and Mathematics Education, University of Oviedo, Campus de Llamaquique, Oviedo 33007, Spain

Published online: 22 January 2025

1 Introduction

Computational Thinking (CT) has emerged as a foundational element in the global educational landscape, transforming our approach to problem-solving, critical thinking, and technological literacy. As technology integration becomes increasingly ubiquitous in diverse societal facets, ranging from professional contexts to daily life, the indispensable role of CT in education cannot be overstated. This paradigm shift extends beyond conventional notions of computer science (CS) education, encapsulating a repertoire of cognitive skills and problem-solving strategies applicable across various disciplines. CT is a fundamental competency that endows individuals with the capacity to think analytically and leverage technology's potential to address a broad spectrum of challenges in our computerized society. It empowers individuals to deconstruct intricate problems, automate processes, and devise complex systems, thereby fostering efficiency and informed decision-making (Wing, 2006).

Even in the early stages of education, it is imperative to introduce children to active engagement with digital technology, cultivating a critical perspective grounded in a profound understanding of what computers can and cannot accomplish for individuals and society at large. The European Commission has actively advocated for the development of CT skills among primary and secondary students in recent years. The Digital Education Action Plan 2021–2027, a component of the European Commission's strategy for a "European Education Area" underscores the significance of quality Computing Education as "one of the priorities to enhance digital skills and competences for the digital transformation" (Bocconi et al., 2022, p. 1). The introduction and integration of CT in the early stages of education have been subjects of study since the term was introduced, underlining its broader scope beyond the use of electronic devices and coding, emphasizing a deep reliance on data. "It is imperative that educators, administrators, and students begin today to consider how to best prepare for and keep pace with this data-driven era of tomorrow" (National Academies of Sciences, Engineering, and Medicine, 2018, p.8). Given the pervasive adoption of Artificial Intelligence (AI), machine learning and deep learning methods in our society, acknowledging the centrality of data in this field is crucial. Indeed, many authors highlight data as the nucleus of the AI-driven revolution (Arteaga Elorriaga, 2023; Bari et al., 2021; Marmanis, 2023). Furthermore, through data literacy, CT establishes a direct connection with mathematical and statistical thinking (see Engel, 2017, and Gleason, 2018). Despite their distinct objectives, they share processes, concepts, and approaches that mutually complement each other (Weintrop et al., 2016). Following these ideas, in this paper we give a comprehensive definition of CT and a thorough analysis of its components, discussing their connection with the curricula.

The purpose of this paper is twofold. Firstly, we scrutinize the current status of CT, with a specific focus on its use in education and its integration into the K-12 curriculum, acknowledging and reflecting its transdisciplinar relation with mathematics due to their synergistic relationship from the educational viewpoint.

Secondly, building upon the accumulation of prior information, we propose a new definition of CT and present a set of components that underscore the significance of data in today's context of data literacy. Additionally, we analyze how this new definition impacts on the CT educational approach, suggesting considering a broader view than reducing CT to programming. The remainder of the paper is organized as follows. Section 2 reviews various definitions of CT, its dimensions, and strategies for implementing CT in K-12 education. In Section 3, we present our new definition of CT and analyze its components. Finally, in Section 4, we discuss the relevance of the new definition along with its implications in the K-12 educational framework. We conclude by highlighting some limitations of the work and pointing out potential avenues for future research.

2 State of the art

2.1 Definitions of CT

Following the groundbreaking work by Papert in the last century (Papert, 1980), numerous authors have contemplated the impact of computing on education. Jeannette Wing has emerged as a prominent author in CT, being the first to associate it with problem-solving skills not exclusive to computer scientists but applicable to everyone (Wing, 2006). In a subsequent definition, Wing proposed that "Computational Thinking is the thought processes involved in formulating problems and their solutions so that the solutions are represented in a form that can be effectively carried out by an information processing agent" (Wing, 2011). Over the past two decades, various authors have grappled with the challenge of gaining insight on this topic.

We have selected systematic and scoping reviews on CT, according to the following criteria: recency (published in 2017 or later), impact (at least 100 citations in the Web of Science database), content relevance (focused on or related to the K-12 educational setting) and content usefulness (addressing or providing information on the CT definitions considered within the reviewed studies). Table 1 presents the selected reviews along with the number of citations across various scientific databases.

The four systematic reviews explore CT from distinct perspectives. Shute et al. (2017) analyzed the components of CT included in various definitions and conceptualizations, proposing a conceptual framework. Hsu et al. (2018) investigated

Table 1 Selected reviews about CT. Source: own elaboration

	Number of citations		
Reference	Web of Science	Scopus	Google Scholar
Shute et al. (2017)	604	796	1519
Hsu et al. (2018)	374	474	838
Tang et al. (2020)	329	368	633
Tikva and Tambouris (2021)	120	141	248

pedagogical strategies and tools for teaching CT in K-12 settings. Tang et al. (2020) examined methods for assessing CT, identifying gaps in evaluation practices, and providing extensive information on K-12 contexts. Tikva and Tambouris (2021) focused on mapping CT through programming within K-12 education. Collectively, these reviews offer a comprehensive overview of the current state of CT in K-12 education. In addition to these reviews, we have considered Román-González et al. (2017), which is not properly a review but includes an interesting overview of CT comprehensive definitions. Lastly, we have also extracted information from a technical report (Bocconi et al., 2022), which is based on Román-González et al.'s classification, because that report deepens into the CT concepts that can be identified in a broad set of CT definitions. The references considered allow us to obtain a broad perspective of the abundance of approaches to defining CT in the literature, this fact underscores the necessity to group and classify the various approaches to this topic. In the following, we are considering the CT definitions included in these reviews, which guarantee we deal with the most noteworthy approaches to the complex nature of CT.

We can discern two primary approaches to defining CT: Programming and/or Problem-solving. It is essential to acknowledge that this categorization is not always crisp, as certain definitions may encompass both approaches. For instance, there are commonalities, such as the significance of breaking down complex problems into smaller components and utilizing algorithms and other techniques to formulate solutions. However, we propose a third approach, wherein the thinker emulates the behavior of a machine, commonly referred to as thinking like a computer. Therefore, in our attempt to provide an overview of the concept of CT in the literature, we expand the classification as follows:

Problem-solving: Some authors focus on the problem-solving facets of computational understanding, advocating the use of computational tools as aids in addressing problem-solving tasks, see Grover et al. (2020), Hazzan et al. (2011), Palts and Pedaste (2020), and Zhang et al. (2021). This emphasis likely stems from the historical development of CT and its close ties with mathematical thinking. In an overarching perspective, Barr and Stephenson (2011) define CT as an approach to solving problems that can be implemented in a computer, while Cansu and Cansu (2019) stress that CT involves breaking down complex problems into more manageable parts, utilizing algorithms and other techniques to develop solutions. Redecker (2017) underscores that this problem-solving approach to CT is vital for success in the digital age, as it can be applied since early childhood education. Within problem-solving definitions, it is noteworthy to consider Wolfram (2020), who argues that the advantages of computation (ubiquity and practicality) enable the development of a wholly computer-based mathematics curriculum using computers. We must stress that the significant role of mathematical problem-solving in this approach might overshadow the computational aspects of CT, possibly depicting it as a subset of mathematical problem solving and detached from fundamental topics in CS.

Programming/Procedural: According to some authors, with Brennan and Resnick (2012) as seminal figures, CT is intricately linked to the algorithmic dimension. They perceive it as a nearly synonymous concept with algorithmic thinking, encompassing components inherent in coding such as debugging, loops, conditionals, etc. (see Israel-Fishelson et al., 2021; Román-Gonzalez et al., 2017; Weintrop, 2018; Witherspoon et al., 2017). In Aho (2012), CT is defined as “the thought processes involved in formulating problems so their solutions can be represented as computational steps and algorithms” (p. 1), closely tied to algorithmic thinking (see Knuth, 1985). Similarly, in Shah (2022), CT is defined as “the process of formulating a problem, finding a solution to the problem, and expressing it in such a way that humans or machines can understand the solution” (p. 2). Extensive literature exists where CT is essentially understood as algorithmic thinking, constraining its dimensions to tasks typically performed during algorithm development or implementation. This perspective may lead to a narrowed view of CT, treating it as a synonym for programming (or even robotics when programs are implemented on robots).

Thinking like a computer: Some researchers place the computer at the core of the thought process, often conceptualizing CT as a way to emulate machine thinking, (see Corradini et al., 2017; Eickelmann et al., 2019; Kafai, 2016; Yaşar, 2018). Denning and Tedre (2019) view CT as a method for interpreting the world: “the mental skills and practices for designing computations that get computers to do jobs for us, and explaining and interpreting the world as a complex of information processes”. The Developing Definition (Selby & Woollard, 2013) also explores various types of thinking involved in CT, such as logical thinking, algorithmic thinking, engineering thinking, and mathematical thinking. According to this definition, CT encompasses a combination of these diverse thinking types.

It is feasible to categorize CT definitions through various lenses. For instance, Tikva and Tambouris (2021) categorizes definitions based on their application domain, while Román-González et al. (2017) classifies CT definitions into three categories, ranging from more generic ones to those specifically focused on problem-solving. Despite the extensive literature on the subject, it becomes evident that there is no universal consensus on the definition of CT. However, the aforementioned classifications underscore that CT represents a distinct way of thinking that goes beyond mere programming or problem-solving abilities.

2.2 Critical analysis of CT dimensions in education

A significant worldwide initiative is underway to equip children with the knowledge necessary to comprehend how computers function and, to some extent, empower children to make computers work. Presently, the term CT finds its place in nearly all secondary schools and many primary school curricula. As we have explored in the preceding subsection, the lack of consensus regarding the definition of CT underscores the apparent ambiguity in its meaning within curricula. This stands in contrast to other knowledge domains, such as mathematics, language, or social sciences, where curricula typically provide clearer specifications. In the realm of CT, curricula often mention the term without explicitly delineating the components they refer to in the K-12 context. The definitions presented in Subsection 2.1 are often too broad for practical application, potentially contributing to the phenomenon of different authors attempting to rephrase each other in an effort to strike a balance between precision and simplicity. Moreover, discrepancies arise concerning the CT components or dimensions depending on the chosen definition.

Various dimensions related to CT are recognized as relevant in educational curricula for primary and secondary education. In Barr and Stephenson (2011), the identified dimensions include abstraction, algorithms and procedures, data collection, data analysis, data representation, parallelization, problem decomposition, and simulation. Selby and Woollard (2013) establish the primary CT elements as abstraction, algorithms, automation, evaluation, problem decomposition, and generalization. In Angeli et al. (2016), the incorporated dimensions are abstraction, decomposition, algorithms, debugging, and generalization. Additionally, Shute et al. (2017) selects abstraction, decomposition, algorithms, debugging, generalization, and iterations as key dimensions associated with CT in the educational context.

In Brennan and Resnick (2012), a programming-centered perspective of CT is presented. The framework divides CT into three key components: computational concepts, encompassing foundational concepts like parallelism, loops, sequences, events, conditionals, operators, and data; computational practices, involving methodologies developed through engagement with these concepts, such as debugging, iteration, reusing and remixing, as well as abstracting and modularizing; and computational perspectives, exploring the cognitive viewpoints that designers cultivate regarding the surrounding world and themselves, including expressing, connecting, and questioning as fundamental cognitive activities.

In Hunsaker (2020), an even broader set of dimensions is categorized into skills (decomposition, pattern recognition, abstraction, algorithm design and evaluation), attitudes (confidence, communication, flexibility), and approaches (tinkering, creating, debugging, persevering, collaborating). The Google Exploring Computational Thinking resources (https://edu.google.com/resources/programs/exploring-computational-thinking/#) gather eleven CT concepts based on ISTE (2022): data organization, data analysis, data representation, abstraction, modeling, simulation, automation, algorithms, debugging, generalization, and transference. These resources were created to provide a better understanding of CT for educators and stakeholders and to support those who want to integrate CT into their own classroom content, teaching practice, and learning.

A comprehensive approach to the conceptualization of CT in European education is presented by Bocconi et al. (2022). The components of CT are divided into a generic problem-solving dimension and an algorithmic or programming dimension (see Table 2).

2.3 The emergence of data

Wing's (2006) conceptualization, from which our literature review stems, frames data as a fundamental component of CT. She views "data as code and code as data" (Wing, 2006, p.33), further relating it to key concepts such as data size, dimension and structure. However, data has gradually been overshadowed in various conceptualizations of CT found in the literature, as we will further analyze in Sect. 4.1.

Table 2 Concepts concerning CT skills development. Bocconi et al. (2022)

CT associated with generic problem-solving	CT associated with programming and computing
Abstraction	Algorithmic Thinking
Data Analysis	Algorithm Design
Data Collection	Automation
Data Representation	Boolean Logic
Decomposition	Computation
Efficiency	Computational Modelling
Evaluation	Conditionals
Generalization	Data Types
Logics & Logical Thinking	Events
Modeling	Functions
Patterns & Pattern Recognition	Iteration
Repeating Patterns	Loops
Simulation	Modularisation
System Thinking	Parallelization
Visualization	Sequencing
	Testing & Debugging
	Threads (Parallel execution)

In a closer look up: The analysis of previous approaches indicates that certain elements of CT are commonly accepted as key components, such as problem-solving and algorithms. While some of these approaches include references to data (see, for example, Barr & Stephenson, 2011), a closer examination reveals that these mentions often lean towards a digital competence perspective (e.g., file management, ethical use of data, graphing, etc.), sometimes missing a deeper focus on data from both Computer Science (CS) and broader Data Science perspectives. Naur (1975) defines data as a representation of facts or ideas in a formalized manner capable of being communicated or manipulated by some process. In CS, data plays a crucial role by enabling machine communication and underpinning the computational ecosystem. While data may initially consist of simple sequences of symbols, their interpretation transforms them into meaningful information. Representations of data are essential not only for computational objectives, such as memory management and program analysis (Lee & Dubovi, 2019), but also for solving problems across disciplines.

In the last decade, researchers and practitioners have emphasized the emergence of data as a transformative factor across all scientific fields, challenging not only methodologies but also epistemologies (Kitchin, 2014, 2021). Data has impacted various areas, including public economic policies and economic research (Einav & Levin, 2014), corporate strategies (Jones & Tonetti, 2020), and finance (Hasan et al., 2020), providing both governments and companies with a powerful tool to support decision-making. The disruption caused by data extends beyond economics to all social sciences, to the point where researchers are discussing the birth of Computational Social Science (Lazer & Ognyanova, 2024). Clearly, data also plays a crucial role in science, as illustrated by its pivotal role in one of the greatest recent advances: genome sequencing (Chen et al., 2021). Furthermore, data is also critical in ongoing developments in other experimental fields such as materials science

(Olivetti et al., 2020), cosmology (Abdalla et al., 2022), and particle physics (Karagiorgi et al., 2022). While data management and representation are foundational to CS, their broader relevance extends to these and many other fields, each utilizing data to address unique challenges and advance knowledge. CS, therefore, serves as an umbrella under which diverse disciplines reside, all sharing a common thread—the utilization, processing, and management of data, as broadly defined earlier. This expanded perspective underscores the centrality of data not only in computational ecosystems but also in the interdisciplinary advancement of knowledge and practice. Wise (2019) stated:

> Recent changes in the size, scope, and types of data available, as well as the kinds of analyses they are subjected to, the ways they move through world, and the purposes to which they are put, create a new ecosystem of data that requires new approaches to education. (Wise, 2019, pp. 1)

The significance of data is particularly paramount in the era of big data, in which the enormous growth of the sources, types and size of generated data explains the importance of the field of Data Science. Its knowledge areas, as defined in the computing curricula of the ACM & IEEE (2021), include (a) computing fundamentals, (b) data acquirement and governance, (c) data management, storage, and retrieval, (d) data privacy, security, and integrity, (e) machine learning, (f) data mining, (g) big data, (h) analysis and presentation. Consequently, younger students should understand the need for different types of data, the importance of the right choice for a particular data-structure, or how the key aspects of computational complexity rely on the size of a dataset, even if the task in hand doesn't involve the actual design of an algorithm. As citizens, they need to develop a basic understanding of topics from disciplines such as Data Science, Artificial Intelligence or Software Engineering, which heavily rely on the concept of extracting knowledge from data. When we look beyond the scope of CS, data emerges as a foundational element that transcends disciplinary boundaries, playing a key role in understanding and addressing problems in various fields. By serving as a pillar of CT, data reinforces its relevance and applicability across a wide range of disciplines, enabling insights and enhancing learning experiences beyond the computational domain.

2.4 CT in K-12 curricula

Computing curricula have been developed worldwide over the last half-century, evolving into several disciplines as computing has permeated society. To some extent, universities have agreed on the core competencies and knowledge that computing undergraduates should develop, even dividing the discipline into subfields such as Computer Science, Information Technology, or Software Engineering (ACM cited as an example). It should be noted in the following paragraphs the contrast between this level of agreement and the variety of approaches in K-12 education in different countries. The lack of tradition to build upon and the non-uniform definition of CT needs to be urgently addressed to develop adequate curricula for these levels with a more homogeneous approach. In this section, we review some of the main aspects of CT curricula across Europe, Australia, India and some Asian-pacific regions.

In Bocconi et al. (2022), an examination is conducted on how various European countries (22 EU Member States and 8 non-EU countries) have incorporated fundamental CS concepts into compulsory education curricula. In this study, CT is conceived (refer to Table 2) both as a concept linked to CS and as a tool to enhance and support a more intricate, discipline-specific, and interdisciplinary approach Jocius et al. (2022). Variations exist among the countries; some emphasize soft skills like critical thinking, creativity, communication, collaboration, personal development, or analytical skills, while others additionally prioritize design and development skills, digital media literacy and awareness, problem-solving, or numeracy skills. The integration of CT skills into curricula also varies among EU countries, but essentially, there are three approaches:

- CT is a cross-curricular theme that should be incorporated into all subjects.
- CT merits dedicated attention in a standalone course.
- CT can be imparted as a component within another course, often integrated into mathematics and occasionally, technology.

In this paper, we will not explore cross-curricular approaches, not because they lack interest, but because, in our view, there is a risk of excessively diluting CT within other subjects. Consequently, the outcome may only marginally manifest the presence of CT. We will go deep into this argument in Subsection 4.2. Therefore, we will pay closer attention to the other two approaches, as they provide a clearer framework for addressing CT about other subjects.

The Australian curriculum (Australian Assessment and Reporting Authority, 2022) identifies two related dimensions, digital technologies knowledge and understanding and digital technologies processes and production skills, that “provide students with knowledge, understanding and skills through which they can safely and ethically exploit the capacity of information systems to systematically transform data into solutions that respond to the needs of individuals, society, the economy and the environment”. The first dimension is focused on the components of digital systems and on the representation of data while the second one is focused on developing skills to create digital solutions to problems and opportunities. Here it includes collecting, managing and analyzing data, defining problems and designing digital solutions, communicating ideas and information. CT arises in this context as a process of recognizing aspects of computation in the world and being able to think logically, algorithmically, recursively and abstractly.

CSpathshala is an ACM India initiative to develop a computing curriculum and initial sample material for K-12 (Shah, 2022). It is remarked in that work that CT can be learned in an unplugged way, stressing that computing concepts and fundamentals do not depend on particular technology or software or programming languages. What is more important, CT fundamentals will stay for much more time than particular technologies. The objectives of this CT curriculum are basically three (Shah, 2022): To acquire an acceptable level of comfort with technology and tools, to use computers for other subjects (like art, mathematics, sciences and geography) and to develop the ability to program a computer to solve a complex task, that involves CT skills like abstraction, decomposition, pattern recognition and solving in incremental steps. Note that the first two objectives are related to digital literacy and only the last one is related to CT.

Regarding Asian Pacific Region, the new national curriculum standards of China include CT as a core literacy of (Liu et al., 2019). The new standard of the senior high school Math curriculum presents some of the skills related to Math as a good starting point to solve problems helped by computers. However, CT is not yet included in the curriculum. CT education in another Asian Pacific Region such as Korea, Taiwan or Hong-Kong is revised in So et al. (2020). Although it seems that CT will be shifted from informal learning to formal programs in school, there is still a lack of pedagogically relevant teaching and learning materials, leading to multiple interpretations of CT core components.

In relation to United States, the country has adopted new CS standards. There are well established Advanced Placement Computer Sciences courses for high school students across the country. However, many states have their own regulations for K-12 and these different state regulations vary considerably (Codelicious, 2024). The model curriculum for K-12 introduced by ACM (Tucker, 2003) established that students must learn to use computers effectively and incorporate the idea of algorithmic thinking as problem solving strategy. In addition, it is highlighted that these skills are only developed in the context of Math education. ACM curriculum is more focused on information technology and digital literacy than on CT.

When discussing a course specifically dedicated to CT, it is essential to consider the existing literature on CS Education. Unlike other cognitive frameworks such as mathematical, abstract, or creative thinking, CT uniquely stands out as denoting both the instructional content and the skill to be fostered. While we do not advocate for a nomenclature change, it is imperative to underscore that fostering CT skills requires introducing students to CS domains, thereby facilitating the emergence of CT skills. For students to grasp an understanding of CS, instructional trajectories in K-12 for CT education should encompass all CT dimensions. Achieving this necessitates an adequate sequencing of these dimensions. In more established educational domains like mathematics, there exists a palpable progression: number sense precedes addition, which in turn is foundational to multiplication; similarly, understanding measure is a precursor to length, this to area, and subsequently leading to volume. CS Education, being in its relative infancy compared to the seasoned domain of Mathematics Education, has considerable explorations ahead to determine the optimal sequencing and depth appropriate for various developmental stages. Illustratively, while a first grader might possess the requisite abstraction skills to address the shortest path problem in a small weighted graph, comprehending the intricacies of a Large Language Model might be beyond his/her ken. Noteworthy initiatives like Bebras (Dagiene & Dolgopolovas, 2022), replete with an extensive collection of age- and dimension-classified unplugged activities, offer a promising avenue for researchers to systematically study the chronological and methodological nuances of CT instruction. As it stands, both theoretical and empirical studies are pivotal to not only validate existing curricular constructs but also to tailor them in alignment with children's cognitive milestones intricately, and also to empirically validate whether the dimensions classified by Bebras align with those obtained in large-scale studies. Regarding the integration of CT as a component in a course dedicated to another subject, as we have mentioned, the most common scenario is its incorporation into mathematics. In this regard, we can identify two approaches: CT for mathematics and mathematics for CT.

Mathematics for CT. Notwithstanding the pronounced interrelationship between CT and Mathematics, for students to authentically cultivate CT skills, there is a need for a direct immersion into these skills. While the mathematical framework offers an endless set of instrumental ideas suitable for elementary programming, the ambit of CT extends into a much broader context demanding comprehensive understanding and engagement. Underlining again the importance of data, we can mention instances such as the storing, sorting, and retrieval of word repositories; the capability to parallelize collaborative tasks; or the systematic indexing of an herbarium. These exemplify challenges that are tailored to foster CT proficiencies, yet they remain detached from a conventional Mathematics curriculum. Consequently, we claim that despite the discernible overlap between mathematical and CT competencies (with both enshrining problem-solving at their core) it is paramount for CT skills to be nurtured through focused and proactive engagement, embracing a focus on competencies that exhibit direct or close transferability, avoiding dependence on peripheral skills that may lack direct transference (Scherer et al., 2019).

CT for Mathematics. In the pursuit of enhancing children's mathematical competence through CT skills development, it is imperative to focus on CT dimensions that exhibit a direct relationship with the targeted mathematical competencies (e.g., Organisation for Economic Co-operation and Development [OECD] (2018). Prevailing scholarly insights suggest that to foster mathematical skills, students should be directly engaged with those specific competencies or ones that are closely related (Boylan et al., 2018; Calao et al., 2015; Laurent et al., 2022). For instance, consider an instructional approach where toddlers are taught to sequence a robot's actions to navigate a designated path to enhance their mathematical skills. At this developmental stage, pivotal mathematical skills include number sense, pattern recognition, and classification. Given the scarce association between these aspects and sequencing tasks, it remains uncertain whether a toddler's programming ability would transfer to improve mathematical skills. While we acknowledge the value of robot-sequencing tasks for young learners, extant literature casts doubt regarding the direct transference of this skill to their mathematical skills (Scherer et al., 2019). On the other hand, thinking computationally on how we represent data such as numbers, polygons or even entity-relationship models, may raise in the classroom very interesting mathematical questions. To name just a few examples: how close can we get to zero in IEEE-754? Why does subtraction become addition in two's complement format? How many different colors can we generate with 8-bits in a simple RGB system? How does a computer decide if a polygon is convex? Or how can we represent inclusive/exclusive characterizations of 2D polygons? All these questions, which may well be introduced as open problems in the math classroom should generate interesting mathematical discussions with the students.

3 A new definition of CT and its components in education

After critically analyzing existing definitions, this section introduces a novel definition of CT. Furthermore, we outline its components, taking into account the educational approach discussed in the preceding section.

3.1 A new definition of CT

Within the analyses presented in Sect. 2, all definitions categorize CT as an umbrella covering different processes, many of them maintaining extensive relationships with mathematical thinking, evident from the profound ties between computer scientists and mathematicians across all facets of CS. It is noteworthy that during the 1970s, myriad mathematicians self-instructed and cultivated their CT abilities ex nihilo (Arbib, 2012), since CS was still being developed at the same time. However, we remark that it is imperative to underline that CT is not merely a subdomain of mathematical thinking but a complementary approach that enables the reciprocal construction of both knowledge domains (Weintrop et al., 2016).

A profound interrelation exists between CT and algorithmic thinking, wherein the crux of computing is anchored in the software development process. Grasping control structures, architecting algorithms, and implementing them across diverse programming languages are quintessential to computing. Yet, CT transcends mere algorithmic design. Analogously, problem-solving stands as a foundational tenet of CT, with the term "problem" being ubiquitous in its definitions (mathematical, statistical, computational, etc.). It is of paramount importance to recognize the awareness of the problem solver regarding the instrumental role computers play within the problem-solving process, endowing her/him with a distinctive vantage point. The third intrinsic facet of any CT process pertains to the data requisite for problem resolution: its domain, dimensions, retrieval, structuring, storage, processing, and analysis. It is somewhat unexpected to observe the limited emphasis this component has garnered within CT definitions, and, subsequently, its timid incorporation into K-12 pedagogical contexts.

Hence, we claim that the three pillars of CT are algorithms, problem- solving and data. While we acknowledge that the majority of CT definitions in Section 2 have adequately addressed the first two pillars, we advocate for the explicit inclusion of the third, namely data. This idea has been already justified in a top-down approach in a data society and will match a bottom-up approach in the description of other components. In pursuit of a succinct definition that underscores the significance of these three pillars in K-12 education, we propose the following:

Computational Thinking (CT) is the way of reasoning that allows people to tackle a problem on some data with the aim of having a computer solve it.[1]

[1] "Pensamiento Computacional (PC) es la manera de razonar que nos permite enfrentarnos a un problema sobre unos datos con el objetivo de que un ordenador lo resuelva." is the suggested translation into Spanish.

While the ideas related to algorithms and problems have always been central to CT, the emphasis on data as a key component in this definition is newer, with only a few references to it, and just as a dimension, that is, second level (see Bocconi et al., 2022). Nonetheless, conceptualizing a computational process devoid of requisite input data, construed in its expansive sense, seems to be impossible, both from a theoretical or research-based point of view and from the practitioner's experience. We add, therefore the data pillar to the CT definition, putting it on the same level as algorithms and problem-solving since, even when no input or output data are explicitly seen in an algorithm, some data, at some point, will be internally processed in every problem.

3.2 CT Components in education

About the multifaceted components that underpin CT, our scrutiny of diverse models is documented in Subsection 2.3. We prefer to avoid the term dimensions as it may imply a factorial structure, which we are not analyzing. Instead, we are solely describing the constituent elements of CT. Despite the absence of a universally accepted description of CT core elements and their granularity, there exist ubiquitous elements across all models (such as abstraction or pattern recognition). Conversely, components like system thinking are less prevalently acknowledged. In this research, we proffer a holistic categorization of CT components grounded in the triadic pillars (algorithms, problems, and data) of the definition we have coined in Subsection 3.1. Our prime ambition is to discern a coherent subset of keywords, competencies, and principles, ensuring an elucidated comprehension of CT's scope in Education, even for those outside the specialized domain.

We have examined several models in Subsection 2.3 regarding the various components that constitute CT. While there is no consensus on the core elements or their level of detail, certain elements are shared among all approaches (such as abstraction or pattern recognition), while others, like system thinking, are less commonly included. In this study, we present a comprehensive classification of CT components based on our three pillars. Our objective is to identify a subset of keywords, skills, and concepts that can provide a more tangible understanding of what CT encompasses, even for non-experts.

Each of the three pillars is characterized by specific keywords or skills, occasionally spanning more than one component. While our goal isn't to pinpoint the precise confines of CT, which would require empirical validation, we aim to offer the educational community a classification of CT components based on their alignment with data, problems, or algorithms.

Figure 1 illustrates the three foundational pillars of our definition of CT along with the associated components. It is worth noting that several components span more than one pillar, prompting our choice for this geometric representation over a mere exhaustive list. The following short description of each component highlights its placement in the diagram:

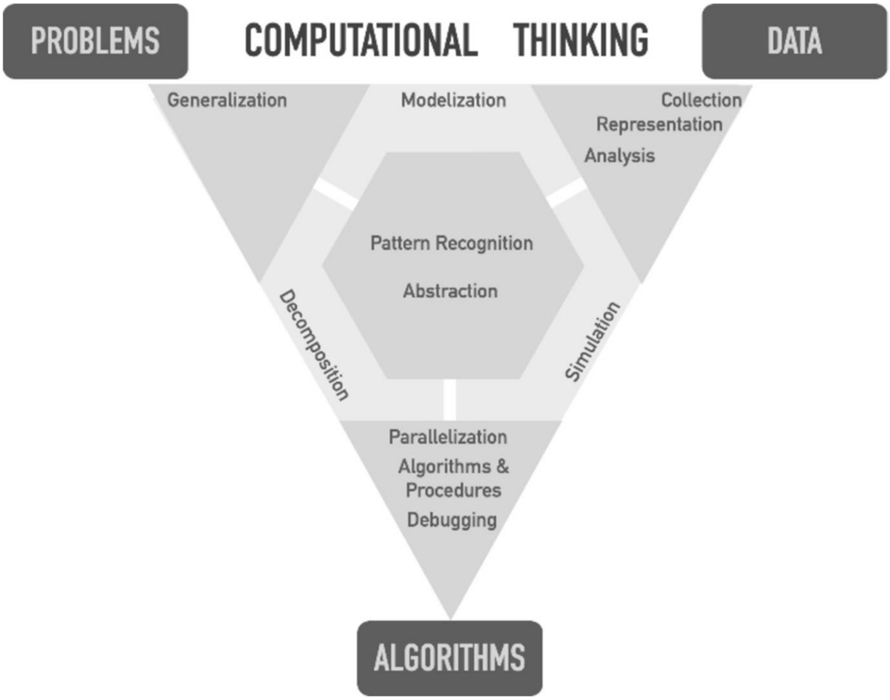

Fig. 1 Components of CT

- Abstraction refers to the process of distilling complex objects, problems, or systems into their essential characteristics, omitting details to facilitate a more manageable understanding, often focusing only on certain details of interest for the purpose of analysis. It is an intellectual operation that involves creating higher-level representations; hence, it is possible to abstract data, algorithms, and problems (Cetin & Dubinsky, 2017).
- Pattern recognition involves the ability to identify and discern recurring structures. It involves the analysis of commonalities in the data, the identification of common problem-solving strategies, and the detection of reoccurring sequences in algorithms. Pattern recognition plays a pivotal role in diverse fields, such as mathematical thinking, image and speech processing, natural language understanding, and machine learning (Ioannidou et al., 2011). In the mathematical educational setting, patter recognition activities are included from very young ages with an orientation to data. Here, we additionally propose to reflect on repetitive actions that might appear in many procedural situations, for example, traditional algorithms for multiplication or addition.
- Modeling encompasses the process of simplifying complex datasets and problem scenarios to enhance analysis, comprehension, and informed decision-making. Through modeling, structured frameworks are developed, assisting in problem-solving and shedding light on the underlying patterns and interrelationships within the data (Magana & Silva Coutinho, 2017). Understanding graphs and

their representation as a data structure is an example on how we reflect on data modeling, in a distinctive perspective from problem modeling.

- Simulation entails the generation of models that replicate or imitate the behavior of datasets and processes. This facilitates the analysis of scenarios that might otherwise be inaccessible or impractical to experience directly. Indeed, it is possible to simulate from hundreds of dice rolls to an entire computer within another computer (Magana & Silva Coutinho, 2017).
- Decomposition refers to the process of breaking down complex problems into smaller, more manageable sub-problems. It involves analyzing the main problem and identifying its constituent parts, which can be tackled independently and then integrated to solve the original problem. Decomposition, which is a fundamental problem-solving technique used to simplify complex tasks, is essential for the design of efficient algorithms (Rich et al., 2018, 2019). Modularity appears naturally in the algorithmic setting even from the early stages of education and is ingrained as well in all mathematical problem-solving tasks.
- Data Collection encompasses the systematic acquisition of relevant and significant data from various sources, such as databases, sensors, records, or files. It includes identifying where to get the data, planning how to collect, extract, or capture it, and then organizing it for further analysis. Understanding the nature of the data, the different data types and the reason for their existence is essential to ensure reliable and high-quality information, which is valuable for analysis, decision-making, and data-driven solutions development in CS.
- Data Analysis involves the systematic examination and interpretation of collected data to uncover meaningful insights and patterns. This encompasses employing statistical and computational techniques to explore, clean, transform, and model the data. The goal of data analysis is to derive valuable information, pinpoint trends, facilitate informed decision-making, and bolster problem-solving processes. Data analysis plays a pivotal role in fields such as machine learning, data mining, and decision-support systems.
- Data Representation concerns the encoding and organization of information in a format conducive to processing and manipulation by computer systems. This not only covers the internal representation of data within a computer's memory but also includes the use of data structures to organize and store data logically. The former, internal data representation, refers to how data is stored and represented at a low level, such as using binary digits (bits) to represent numbers and characters. Conversely, data structures are higher-level constructs that facilitate efficient data organization and management. These structures elucidate the interrelations and arrangement of data components, enabling operations such as search, sort, and retrieval. Effective data representation and selection of appropriate data structures are paramount for efficient data processing and optimizing computational operations. At lower levels in education, a general insight on binary representation and internal behavior of binary arithmetic may raise awareness on bigger-scale issues for large amounts of data.
- Algorithms and Procedures serve as foundational concepts in the design and development of efficient, organized solutions. They are quintessential to problem-solving skills and foster the formulation of scalable, well-structured

software solutions. This approach entails breaking down a larger problem into smaller, independent modules that can be developed and tested separately, thereby enhancing reusability and maintainability (Stephens & Kadijevich, 2020).
- Parallelization involves the simultaneous, concurrent execution of multiple tasks or processes to enhance computational efficiency. It enables us to harness the power of parallel processing, where computations are distributed across multiple computing units, such as processors or threads. Parallelization not only speeds up algorithm execution but also manages large-scale data more effectively. The parallelization is intrinsically linked with a thorough examination of algorithmic complexity and a reflective view of the underlying data structures (Kirkpatrick, 2017). Most block-based languages designed specifically for education are based on distributed computing, allowing a natural approach to this concept.
- Debugging stands as fundamental concepts underpinning the reliability and correctness of software systems and algorithm implementation. Debugging is the process of identifying, isolating, and fixing errors or bugs in the code. This aspect accentuates the importance of software's accuracy and functionality, and code quality. Testing, which is part of the debugging process, systematically verifies the functionality and performance of programs to unearth errors, inconsistencies, or unexpected behaviors. Through comprehensive test scenarios, we can affirm the software's accuracy and validate its intended functionality (Wong & Jiang, 2018).
- Generalization denotes the ability to apply knowledge, concepts, and skills learned in one context to varied and diverse scenarios, extracting broader insights from instances, and leveraging these insights for universal application. This entails grasping underlying principles and patterns to address a wide range of problems transcending the explicit instances experienced during the learning phase. Through the lens of generalization, the aim is to transfer understanding and expertise to novel, unfamiliar situations, thereby enhancing adaptability and creativity in problem-solving. Note that the term generalization is understood with different nuances in mathematics and in CT (van Borkulo et al., 2021).

4 Discussion and conclusions

Drawing from the suggested definition of CT and its foundational principles, this section delves into its significance and consequences within the K-12 educational context. First, we review what our definition adds to previous work on this issue. We then examine its compatibility with several K-12 curricula and analyze how the inclusive components can inform the development of curricula aimed at nurturing CT skills among students. Additionally, we highlight the importance of extensive research into students' learning paths and age-appropriate concepts, presenting potential avenues for future research in this field.

4.1 What does our definition contribute?

In this paper, we have introduced a novel definition of CT based on three key elements, among which data gains particular relevance and emerges as a crucial concept in CT. The Big Data era is characterized by an unprecedented increase in data volume, speed, and diversity. Our definition consequently places data, together with algorithms and problems, at the heart of the definition of CT. Regarding the set of components in our approach, all selected elements were already mentioned in previous descriptions and definitions of CT, and our presentation uniquely situates them in a hierarchical relationship to the three main pillars: problems, data and algorithms. Moreover, we underscore the importance of integrating data-specific components (including its collection, organization, cleaning, storage, processing, etc.) within the foundational components of CT.

We have analyzed and contrasted five of the most relevant papers in terms of citations in scientific literature, in chronological order: Wing (2006), Barr and Stephenson (2011), Brennan and Resnick (2012), Selby and Woollard (2013), and Angeli et al. (2016). In all of them the authors propose a definition of CT and a list of elements, keywords or topics associated to CT. Even though the authors explain what they understand with "thinking computationally", only two of them (Selby & Woollard, 2013; Angeli et al., 2016) provide a formal discussion on the list of chosen components. A more informal approach, probably due to the absence of previous literature on the issue, is given by the seminal papers of Wing, Barr and Stephenson and Brennan and Resnick.

Figure 2 summarizes how each paper refers to each of the components that we have selected in our model. Our graph shows "Yes" (light gray shaded components) if there is an explicit mention to that component as an essential element in that paper; we use "Partially/Implicit" (gray shade) when the element appears implicitly, partially referenced or just as an example in a list; we use "No" (dark gray shade) for

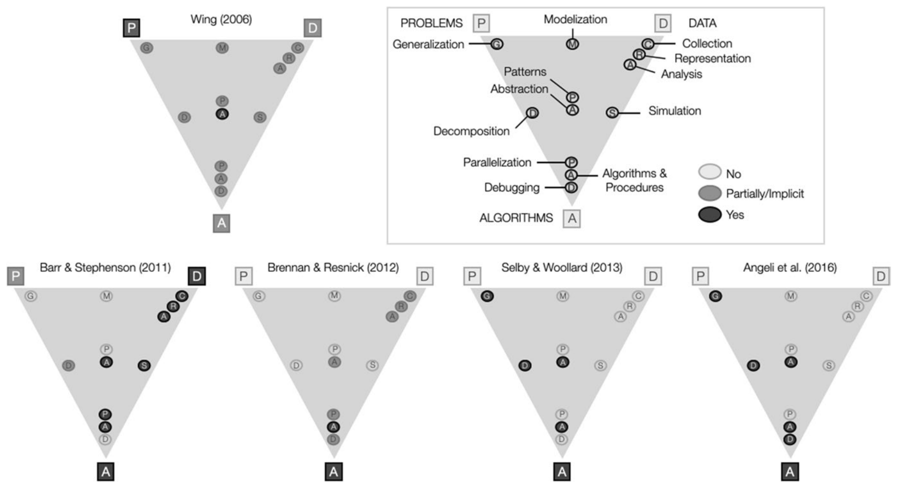

Fig. 2 Components of CT in most cited authors

the remaining cases. A closer look at the graph in the Figure allows us to represent each component and travel chronologically starting in Wing's paper, where she mentions dozens of concepts related to how problems are faced in Computer Science or, in her own words, "Thinking like a computer scientist". The extensive list that Wing proposes includes all our selected components, with abstraction and problems being explicitly mentioned in the definition. Despite Wing's focus on problems, the focus on the problem-solving approach moves in the literature towards the pillar of algorithms, which appears explicitly in all four selected papers. Despite Barr and Stephenson's explicit reference to the data component, these elements have faded away in the remaining works, where the algorithmic component seems to be gaining weight up to the point where CT and Algorithmic Thinking are at risk of taking the part for the whole and becoming synonyms.

The other most novel aspect of the new definition is the detailed analysis of what we have termed components (and, later in the conclusions, we will explain why this terminological choice). Not only is a comprehensive list of terms describing skills, competencies, abilities, dimensions, or characteristics of CT elaborated, but also a detailed description is provided on how to interpret each of these components. This is essential as terms such as abstraction, generalization or modeling may have different nuances (even, different meanings) when analyzed from the standpoint of mathematical thinking versus CT. Additionally, as evident in Fig. 2, the new definition provides an integrated and hierarchical view of the components that have already appeared partially in previous definitions.

4.2 A new approach to CT in K-12 curricula

The new definition and characterization of CT broadens the focus by placing greater emphasis on data, making it novel compared to previous approaches that have focused more on programming and algorithms or problem-solving. While algorithms and problem-solving are also present in our model, they are not exclusive. In our view, this redefinition of CT in the educational field should have implications for how CT is conceptually addressed in the curriculum and, consequently, in the methods used for its teaching and learning.

It is pivotal to recognize that official curriculum guidelines in different countries not only guide teachers but also influence textbook content and delineate assessment standards (Coles et al., 2023). This fact is substantial upon curriculum designers, especially when pioneering the foundational structure of an emergent subject area such as CT. This responsibility is accentuated given that the few research about CT teacher training revealed that in-service teacher training programs tend to focus intensively on the conceptual content of computer science, neglecting its pedagogy and its integration into the classroom (Mason & Rich, 2019). This focus on computational aspects weakens the formal training in both the pedagogical methods and the substantive content of newer fields such as CT (Hubbard, 2018; Sands et al., 2018). On the other hand, teachers tend to have low levels of perceived readiness for teaching CT (Santaengracia et al., 2023). Thus, our aim is not only redefining CT but also to contextualize our model in K-12 education, illustrating how this allows us to move beyond the association of CT teaching solely with programming, responding to Stephens' claim (2018) about the need of a broader consideration of the curriculum that embraces the concept of CT and paves the way for an effective classroom implementation.

There is a lack of consensus regarding the integration of CT into different educational systems. As mentioned in Subsection 2.3, the incorporation of CT skills into curricula varies across EU countries, with three primary approaches: CT as a cross-curricular theme integrated into all subjects, CT offered as an independent course, and CT integrated as a component of another course. In Subsection 2.3, we have thoroughly analyzed the approaches we have labeled Mathematics for CT and CT for Mathematics. Beyond these interrelations and dependencies between CT and mathematics, the new definition of CT necessitates a reconsideration of the existing relationship between programming (including algorithms) and CT, highlighting that CT should not be confined to programming alone. The prevalent, yet somewhat limited, view of CT as synonymous with programming jeopardizes the fundamental notion that CS is an independent science rather than merely a tool to facilitate mathematics learning. Embedding CT in mathematics also opens avenues such as Wolfram's proposition to teach mathematics extensively using computers (Wolfram, 2020). In our discourse, we spotlight the cultivation of the specific components discussed in Subsection 2.2 aiming at providing a coherent discourse for CT education.

Having acknowledged that skills are hard to transfer, we understand that each CT component is best nurtured either in isolation or within closely related contexts. We have to discern thus which components we want to foster within the K-12 educational spectrum and the rationale behind such choices. To achieve this, we must ground our decisions in the pillars and components defined in Section 3.2 and allocate specific curricular segments dedicated to the development of these CT skills. Identifying the three pillars of the definition helps us distinguish between algorithmic and CT and opens up to understanding how data are collected, represented and analyzed using a problem-solving educational approach. Furthermore, identifying data as the third pillar in the definition of CT not only contributes to acknowledging its role as an information provider to the other two pillars (problems and algorithms) but also strengthens the connection with aspects of the so-called data literacy (OECD, 2019). Data literacy is related to both mathematical and statistical thinking. While traditionally included within the realm of mathematics in the European academic tradition, it differs in certain aspects from mathematical thinking, a distinction not always emphasized in the North American tradition (see, for instance, Diggle, 2015; Rodríguez-Muñiz et al., 2022). In this way, our definition of CT establishes a closer link with other forms of thinking, such as statistical thinking, enabling the development of stochastic reasoning (Batanero, 2019). Consequently, it establishes a direct connection between CT and Data Science education (Engel, 2017).

4.3 Conclusions

Even today, the definition of CT remains a subject of ongoing scholarly debate. The components, skills, or dimensions attributed to CT are softly outlined, contingent upon one's interpretation of the concept, yet misconceptions persist. It is noteworthy to differentiate between digital competence and CT; while they may occasionally be conflated, they represent distinct domains, as we stated in Section 1. Furthermore, CT should not be misinterpreted as merely a subset of mathematics or mathematical thinking, nor as a mere tool for mathematical algorithmic programming. Such misinterpretations can inadvertently result in a skewed educational focus on CT, disproportionately emphasizing dimensions closely aligned with mathematics at the expense of a comprehensive understanding of CT, and, by extension, CS.

The definition introduced in this paper, with three pillars named algorithms, problem-solving, and data, together with the analysis of its components provides a clear framework to understand what mathematical thinking and CT have in common and how they differ. It assumes that CT goes way beyond programming and establishes the foundations for a K-12 education on coherent CT, aligning with this model and the approaches that have been developed under various paradigms in different countries. In K-12, CT should be cultivated independently, given its foundational role in CS, analogous to the centrality of mathematical thinking in mathematics. In the current era, honing CS skills is imperative, making comprehensive CT education integral to any curriculum. However, vigilance is required to ensure that CT is not solely equated with algorithmic thinking or with problem-solving, which would constrict the more expansive understanding of what CS truly encompasses.

It is anticipated that there will be an increased emphasis on CT in initial teacher training programs, especially for generalist teachers, as is the case in many countries for primary education (European Education and Culture Executive & Agency, 2015). In such contexts, the integration of CT may be more straightforward, as the interdisciplinary nature of the training facilitates its inclusion, for instance, linked to STEAM initiatives. For specialized teacher training programs, common in many countries for secondary education (Muñiz-Rodríguez et al., 2016), this integration could be more challenging. Its success would hinge on the subject or specialty with which CT is primarily associated (for instance, mathematics in the case of Spain).

It is important to point out the limitations of this study. Firstly, we have employed an approach based on the review of literature and the analysis of computational activity to provide a definition of CT. Consequently, the resulting definition and dimensions model necessitates validation by experts, a process we are already initiating through a Delphi method, which will be part of a forthcoming paper. Secondly, the model would also require empirical validation. Deliberately, we have referred to pillars and components of CT rather than dimensions or another term that might suggest a factorial structure. The description of components and our proximity model to the three pillars in Fig. 1 requires empirical validation based on appropriate instruments. In this regard, the interrelation between some of the components we have identified is evident. Moreover, the scarce empirical studies conducted when analyzing CT domains beyond programming/algorithmic aspects have revealed that the factorial structure is simpler than it may appear. Thus, the so-called dimensions might actually be facets or nuances within the same factor. Therefore, the design of analysis instruments should be approached following the pillars and components of our definition to shed light on the factorial structure of the model. As an example, Bebras (Dagiene & Sentance, 2016) initiative distinguishes several dimensions, however, this has been questioned, especially if these are understood as cognitive factors: only three factors configuring CT have been identified (Souto Oliveira et al., 2021).

To conclude, we must underline the need for the emergence and development of formal didactics of CT, akin to that of other disciplines such as mathematics education. This entails research and analysis into how CT is taught and learned, aiming to establish it as a distinct entity in fundamental research. This perspective extends beyond classroom practice while maintaining a vital connection to it, incorporating approaches like design-based research. Building the didactics of CT would enable a more systematic and structured development of educational strategies, thereby contributing to CT effective training.

Acknowledgements This research has been partially funded by:

• JJS: The Ministry of Universities of Spain (Grant FPU21/05874), by Ministerio de Ciencia e Innovación of Spain (Project PID2022-139886NB-I00).

• ID: The project with code PID2022-139886NB-I00 by Ministerio de Ciencia e Innovación of Spain.

• JJS and LJRM: The project PID2021-122180OB-100 by Ministerio de Ciencia e Innovación of Spain. Authors belong to MTSK network.

Author contributions Belén Palop: Conceptualization, Methodology, Software, Investigation, Resources, Supervision, Writing - Original Draft, Writing – review & editing, Project administration. Irene Díaz: Conceptualization, Methodology, Software, Investigation, Resources, Supervision, Writing - Original Draft, Writing – review & editing. Luis J. Rodríguez-Muñiz: Conceptualization, Methodology, Investigation, Resources, Supervision, Writing – review & editing. Juan José Santaengracia: Conceptualization, Methodology, Software, Investigation, Resources, Supervision, Writing - Original Draft, Writing – review & editing.

Funding Open Access funding provided thanks to the CRUE-CSIC agreement with Springer Nature.

Data availability No datasets were generated or analyzed during the current study.

Declarations

Competing interest Authors do not declare any competing interests.

References

Abdalla, E., Abellán, G. F., Aboubrahim, A., Agnello, A., Akarsu, Ö., Akrami, Y., Alestas, G., Aloni, D., Amendola, L., Anchordoqui, L. A., Anderson, R. I., Arendse, N., Asgari, M., Ballardini, M., Barger, V., Basilakos, S., Batistca, R. C., Battistelli, E. S., Battye, R., & Pettorino, V. (2022). Cosmology intertwined: A review of the particle physics, astrophysics, and cosmology associated with the cosmological tensions and anomalies. *Journal of High Energy Astrophysics*, *34*, 49–211. https://doi.org/10.1016/j.jheap.2022.04.002

ACM & IEEE. (2021). *Computing Curricula 2020 CC2020 paradigms for global Computing Education*. ACM & IEEE.

Aho, A. V. (2012). Computation and computational thinking. *The Computer Journal*, *55*(7), 832–835. https://doi.org/10.1093/comjnl/bxs074

Angeli, C., Voogt, J., Fluck, A., Webb, M., Cox, M., Malyn-Smith, J., & Zagami, J. (2016). A K-6 computational thinking Curriculum Framework: Implications for teacher knowledge. *Journal of Educational Technology & Society*, *19*(3), 47–57.

Arbib, M. A. (2012). *Brains, Machines, and Mathematics*. Springer Science & Business Media.

Arteaga Elorriaga, A. (2023). The importance of data integrity in the era of big data, artificial intelligence and digital hyperconnection. *Pertsonak eta Antolakunde Publikoak Kudeatzeko Euskal Aldizkaria*, *5*, 30–51. https://doi.org/10.47623/ivaprvgp.06.2023.AB.02

Australian Assessment and Reporting Authority (2022). Structure | The Australian curriculum (Version 8.4). https://www.australiancurriculum.edu.au/f-10-curriculum/technologies/digital-technologies/structure/

Bari, V., Gaikwad, D., & Babar, D. R. (2021). A review of machine learning and its applications. *International Journal of Engineering Applied Sciences and Technology*, *6*(3), 190–193. https://doi.org/10.33564/IJEAST.2021.v06i03.029

Barr, V., & Stephenson, C. (2011). Bringing computational thinking to K-12: What is involved and what is the role of the computer science education community? *ACM Inroads*, *2*(1), 48–54. https://doi.org/10.1145/1929887.1929905

Batanero, C. (2019). Statistical sense in the information society. In K. O. Villalba, A. Adúriz, F. J. García & J. Lavonen (Eds.), *Proceedings of the Congreso Internacional Sobre Educación y Tecnología en Ciencias* (pp 28–38). CEUR-WS.

Bocconi, S., Chioccariello, A., Kampylis, P., Dagienė, V., Wastiau, P., Engelhardt, K., Earp, J., Horvath, M. A., Jasutė, E., Malagoli, C., Masiulionytė-Dagienė, V., & Stupurienė, G. (2022). March). *Reviewing Computational Thinking in Compulsory Education*. Joint Research Centre (JRC). https://doi.org/10.2760/126955

Boylan, M., Demack, S., Wolstenholme, C., Reidy, J., & Reaney, S. (2018). *ScratchMaths: Evaluation report and executive summary*. *Education Endowment Foundation*. https://shura.shu.ac.uk/23758/1/ScratchMaths%20evaluation%20report.pdf

Brennan, K., & Resnick, M. (2012). *New frameworks for studying and assessing the development of computational thinking*. Annual American Educational Research Association meeting.

Calao, L. A., Moreno-León, J., Correa, H. E., & Robles, G. (2015). Developing mathematical thinking with scratch. In G. Conole, T. Klobučar, C. Rensing, J. Konert & E. Lavoué (Eds.), *Design for teaching and learning in a networked world* (pp. 17–27). Springer International Publishing. https://doi.org/10.1007/978-3-319-24258-3_2

Cansu, F. K., & Cansu, S. K. (2019). An overview of computational thinking. *International Journal of Computer Science Education in Schools*, *3*(1), 17–30. https://doi.org/10.21585/ijcses.v3i1.53

Cetin, I., & Dubinsky, E. (2017). Reflective abstraction in computational thinking. *The Journal of Mathematical Behavior*, *47*, 70–80. https://doi.org/10.1016/j.jmathb.2017.06.004

Chen, T., Chen, X., Zhang, S., Zhu, J., Tang, B., Wang, A., Dong, L., Zhang, Z., Yu, C., Sun, Y., Chi, L., Chen, H., Zhai, S., Sun, Y., Lan, L., Zhang, X., Xiao, J., Bao, Y., Wang, Y., & Zhao, W. (2021). The genome sequence Archive Family: Toward Explosive Data Growth and Diverse Data types. *Genomics Proteomics & Bioinformatics*, *19*(4), 578–583. https://doi.org/10.1016/j.gpb.2021.08.001

Codelicious (2024). United States K-12 computer science standards. https://www.codelicious.com/united-states. Accessed Jan 2024.

Coles, A., Rodríguez-Muñiz, L. J., Mok, I. A. C., Ruiz, Á., Karsenty, R., Martignone, F., Osta, I., Ferretti, F., & Nguyen, T. T. A. (2023). Teachers, resources, Assessment practices: Role and Impact on the curricular implementation process. In Y. Shimizu, & R. Vithal (Eds.), *Mathematics Curriculum reforms around the World: The 24th ICMI Study* (pp. 291–321). Springer International Publishing. https://doi.org/10.1007/978-3-031-13548-4_18

Corradini, I., Lodi, M., & Nardelli, E. (2017). Conceptions and misconceptions about computational thinking among Italian primary school teachers. *Proceedings of the 2017 ACM Conference on*

International Computing Education Research (pp. 136–144). ACM. https://doi.org/10.1145/3105726.3106194

Dagiene, V., & Dolgopolovas, V. (2022). Short tasks for scaffolding computational thinking by the global Bebras Challenge. *Mathematics*, *10*(17), 3194. https://doi.org/10.3390/math10173194

Dagiene, V., & Sentance, S. (2016). It's computational thinking! Bebras tasks in the curriculum. *9973*. https://doi.org/10.1007/978-3-319-46747-4_3

Denning, P. J., & Tedre, M. (2019). *Computational thinking*. MIT Press.

Diggle, P. J. (2015). Statistics: A Data Science for the 21st Century. *Journal of the Royal Statistical Society Series A: Statistics in Society*, *178*(4), 793–813. https://doi.org/10.1111/rssa.12132

Eickelmann, B., Labusch, A., & Vennemann, M. (2019). Computational thinking and problem-solving in the context of IEA-ICILS 2018. In D. Passey, R. Bottino, C. Lewin & E. Sanchez (Eds.), *Empowering Learners for Life in the Digital Age* (pp. 14–23). Springer International Publishing. https://doi.org/10.1007/978-3-030-23513-0_2

Einav, L., & Levin, J. (2014). The data revolution and economic analysis. *Innovation Policy and the Economy*, *14*(1), 1–24. https://doi.org/10.1086/674019

Engel, J. (2017). Statistical literacy for active citizenship: A call for data science education. *Statistics Education Research Journal*, *16*(1), 44–49. https://doi.org/10.52041/serj.v16i1.213

European Education and Culture Executive, & Agency, E. (2015). *The teaching profession in Europe: Practices, perceptions, and policies*. Publications Office of the European Union. https://data.europa.eu/doi/10.2797/031792

Gleason, N. W. (Ed.). (2018). Higher education in the era of the fourth industrial revolution. Springer Nature. https://doi.org/10.1007/978-981-13-0194-0

Grover, S., Fisler, K., Lee, I., & Yadav, A. (2020). Integrating computing and computational thinking into K-12 STEM learning. *Proceedings of the 51st ACM Technical Symposium on Computer Science Education*, 481–482. https://doi.org/10.1145/3328778.3366970

Hasan, M. M., Popp, J., & Oláh, J. (2020). Current landscape and influence of big data on finance. *Journal of Big Data*, *7*(21), 1–17. https://doi.org/10.1186/s40537-020-00291-z

Hazzan, O., Lapidot, T., & Ragonis, N. (2011). *Guide to teaching computer science: An activity-based approach*. Springer London. https://doi.org/10.1007/978-0-85729-443-2

Hsu, T. C., Chang, S. C., & Hung, Y. T. (2018). How to learn and how to teach computational thinking: Suggestions based on a review of the literature. *Computers & Education*, *126*, 296–310. https://doi.org/10.1016/j.compedu.2018.07.004

Hubbard, A. (2018). Pedagogical content knowledge in computing education: A review of the research literature. *Computer Science Education*, *28*(2), 117–135. https://doi.org/10.1080/08993408.2018.1509580

Hunsaker, E. (2020). Computational thinking. In A. Ottenbreit-Leftwich, & R. Kimmons (Eds.), *The K-12 Educational Technology Handbook* (pp. 91–109). EdTech Books.

Ioannidou, A., Bennett, V., Repenning, A., Koh, K. H., & Basawapatna, A. (2011). Computational thinking patterns. Paper presented at the annual meeting of the American educational research association. https://files.eric.ed.gov/fulltext/ED520742.pdf. Accessed Jun 2022.

Israel-Fishelson, R., Hershkovitz, A., Eguíluz, A., Garaizar, P., & Guenaga, M. (2021). The associations between Computational thinking and Creativity: The role of personal characteristics. *Journal of Educational Computing Research*, *58*(8), 1415–1447. https://doi.org/10.1177/0735633120940954

ISTE (2022). *ISTE Computational Thinking Competencies.* https://iste.org/standards/computational-thinking competencies*. Accessed Jan 2024.*

Jocius, R., O'Byrne, W. I., Albert, J., Joshi, D., Blanton, M., Robinson, R., Andrews, A., Barnes, T., & Catete, V. (2022). Building a virtual community of practice: Teacher learning for computational thinking infusion. *TechTrends*, *66*(3), 547–559. https://doi.org/10.1007/s11528-022-00729-6

Jones, C. I., & Tonetti, C. (2020). Nonrivalry and the Economics of Data. *American Economic Review*, *110*(9), 2819–2858. https://doi.org/10.1257/aer.20191330

Kafai, Y. (2016). From computational thinking to computational participation in K-12 education. *Communications of the ACM*, *59*(8), 26–26. https://doi.org/10.1145/2955114

Karagiorgi, G., Kasieczka, G., Kravitz, S., Nacham, B., & Shih, D. (2022). Machine learning in the search for new fundamental physics. *Nature Reviews Physics*, *4*, 399–412. https://doi.org/10.1038/s42254-022-00455-1

Kirkpatrick, K. (2017). Parallel computational thinking. *Communications of the ACM*, *60*(12), 17–19. https://doi.org/10.1145/3148760

Kitchin, R. (2014). Big Data, new epistemologies and paradigm shifts. *Big Data & Society*, *1*(1). https://doi.org/10.1177/2053951714528481

Kitchin, R. (2021). *Data lives: How data are made and shape our world*. Bristol University.
Knuth, D. E. (1985). Algorithmic thinking and Mathematical. *The American Mathematical Monthly*, *92*(3), 170–181. https://doi.org/10.1080/00029890.1985.11971572
Laurent, M., Crisci, R., Bressoux, P., Chaachoua, H., Nurra, C., de Vries, E., & Tchounikine, P. (2022). Impact of programming on primary mathematics learning. *Learning and Instruction*, *82*, 101667. https://doi.org/10.1016/j.learninstruc.2022.101667
Lazer, D., & Ognyanova, K. (2024). The future of computational social science. In J. M. Box-Steffensmeier, D. P. Christenson & V. Sinclair-Chapman (Eds.), Oxford Handbook of Engaged Methodological Pluralism in Political Science (online edition). Oxford Academic. https://doi.org/10.1093/oxfordhb/9780192868282.013.50
Lee, V. R., & Dubovi, I. (2019). At home with data: Family engagements with data involved in type 1 *Diabetes Management Journal of the Learning Sciences*, 29(1), 11–31. https://doi.org/10.1080/10508406.2019.1666011.
Liu, Y., Ma, Z., & Qian, Y. (2019). Developing Chinese elementary school students' computational thinking: A convergent cognition perspective. In V. Chopella & D.B. Phatak (Eds.) Proceedings of the ACM Conference on Global Computing Education (pp. 238–238). https://doi.org/10.1145/3300115.3312514
Magana, A. J., & Silva Coutinho, G. (2017). Modeling and simulation practices for a computational thinking-enabled engineering workforce. *Computer Applications in Engineering Education*, *25*(1), 62–78. https://doi.org/10.1002/cae.21779
Marmanis, H. (2023). *The importance of data for AI.* https://www.copyright.com/blog/the-importance-of-data-for-ai/. Accessed 11-11-2023.
Mason, S. L., & Rich, P. J (2019). Preparing elementary school teachers to teach computing, coding, and computational thinking. *Contemporary Issues in Technology and Teacher Education*, *19*(4), 790–824.
Muñiz-Rodríguez, L., Alonso Velázquez, P., Rodríguez-Muñiz, L. J., & Valcke, M. (2016). Is there a gap in initial secondary mathematics teacher education in Spain compared to other countries? *Revista De Educación*, *372*, 111–140. https://doi.org/10.4438/1988-592X-RE-2015-372-317
National Academies of Sciences, Engineering, and Medicine. (2018). *Data science for undergraduates: Opportunities and options*. National Academies.
Naur, P. (1975). Concise survey of computer methods. Petrocelli. http://www.naur.com/Conc.Surv.html. Accessed Jan 2024.
OECD (2019). An OECD learning framework 2030. In G. Bast, E. G. Carayannis, & D. F. J. Campbell (Eds.), *The Future of Education and Labor* (pp. 23–35). Springer International Publishing. https://doi.org/10.1007/978-3-030-26068-2_3
Olivetti, E. A., Cole, J. M., Kim, E., Kononova, O., Ceder, G., Han, T. Y. J., & Hiszpanski, A. M. (2020). Data-driven materials research enabled by natural language processing and information extraction. *Applied Physics Reviews*, *7*(4), 041317. https://doi.org/10.1063/5.0021106
Organisation for economic co-operation and development [OECD]. (2018, September). *PISA for Development Assessment and Analytical Framework: Reading, Mathematics and Science*. OECD Publishing. https://doi.org/10.1787/9789264305274-en
Palts, T., & Pedaste, M. (2020). A model for developing computational thinking skills. *Informatics in Education*, *19*(1), 113–128. https://doi.org/10.15388/INFEDU.2020.06
Papert, S. (1980). *Mindstorms: Children, computers, and powerful ideas*. Basic Books.
Redecker, C. (2017). European framework for the digital competence of educators: DigCompEdu. In Y. Punie (Ed.), *Digital competence of educators*. Publications Office of the European Union.
Rich, K. M., Binkowski, T. A., Strickland, C., & Franklin, D. (2018). Decomposition: A K-8 computational thinking learning trajectory. *Proceedings of the 2018 ACM Conference on International Computing Education Research*, 124–132. https://doi.org/10.1145/3230977.3230979
Rich, P. J., Egan, G., & Ellsworth, J. (2019). A framework for decomposition in computational thinking. *Proceedings of the 2019 ACM Conference on Innovation and Technology in Computer Science Education*, 416–421. https://doi.org/10.1145/3304221.3319793
Rodríguez-Muñiz, L. J., Muñiz-Rodríguez, L., García-Alonso, I., López-Serentill, P., Vásquez, C., & Alsina, À. (2022). Navigating between abstraction and context in secondary school statistics education. *Culture and Education*, *34*(3), 689–725. https://doi.org/10.1080/11356405.2022.2058794
Román-Gonzalez, M., Perez-Gonzalez, J. C., & Jimenez-Fernandez, C. (2017). Which cognitive abilities underlie computational thinking? Criterion validity of the computational thinking test. *Computers in Human Behavior*, *72*(100), 678–691. https://doi.org/10.1016/j.chb.2016.08.047
Román-González, M., Moreno-León, J., & Robles, G. (2017). Complementary tools for computational thinking assessment. In S. C. Kong, J. Sheldon, & K. Y. Li (Eds.), *Proceedings of International*

Conference on Computational Thinking Education (CTE 2017) (pp 154–159). The Education University of Hong Kong.
Sands, P., Yadav, A., & Good, J. (2018). *Computational thinking in K-12: In-service teacher perceptions of computational thinking*. Springer.
Santaengracia, J. J., Palob, B., & Rodríguez-Muñiz, L. J. (2023). Percepciones Del Profesorado Sobre Pensamiento Computacional. Estudio De Una Formación. In C. Jiménez-Gestal, Á. Magreñán, E. Badillo, & P. Ivars (Eds.), *Investigación en Educación Matemática XXVI* (pp. 491–498). SEIEM.
Scherer, R., Siddiq, F., & Sánchez Viveros, B. (2019). The cognitive benefits of learning computer programming: A meta-analysis of transfer. *Journal of Educational Psychology*, *111*, 764–792. https://doi.org/10.1037/edu0000314
Selby, C. C., & Woollard, J. (2013). *Computational Thinking: The Developing Definition*. https://core.ac.uk/download/pdf/17189251.pdf. Accessed Jan 2024.
Shah, V. (2022). *CSpathshala: Computational Thinking Curriculum for K-12*. ACM India. Shimizu, Y., & Vithal, R. (Eds.). (2023). *Mathematics Curriculum Reforms Around the World: The 24th ICMI Study*. Springer International Publishing. https://doi.org/10.1007/978-3-031-13548-4
Shute, V. J., Sun, C., & Asbell-Clarke, J. (2017). Demystifying computational thinking. *Educational Research Review*, *22*, 142–158. https://doi.org/10.1016/j.edurev.2017.09.003
So, H. J., Jong, M. S. Y., & Liu, C. C. (2020). Computational thinking education in the Asian Pacific Region. *The Asia-Pacific Education Researcher*, *29*(1), 1–8. https://doi.org/10.1007/s40299-019-00494-w
Souto Oliveira, A. L., Andrade, W. L., Guerrero, S., D. D., & Araujo Melo, M. R. (2021). How do Bebras tasks explore algorithmic thinking skill in a computational thinking contest? *2021 IEEE Frontiers in Education Conference (FIE)*, 1–7. https://doi.org/10.1109/FIE49875.2021.9637151
Stephens, M. (2018). Embedding algorithmic thinking more clearly in the mathematics curriculum. In Y. Shimizy & R. Vithal (Eds.), *ICMI study 24 conference proceedings. School mathematics curriculum reforms: Challenges, changes and opportunities* (pp. 483–490). ICMI & Tsukuba University.
Stephens, M., & Kadijevich, D. M. (2020). Computational/algorithmic thinking. In S. Lerman (Ed.), *Encyclopedia of Mathematics Education* (pp. 117–123). Springer International Publishing. https://doi.org/10.1007/978-3-030-15789-0_100044
Tang, X., Yin, Y., Lin, Q., Hadad, R., & Zhai, X. (2020). Assessing computational thinking: A systematic review of empirical studies. *Computers & Education*, *148*, 103798. https://doi.org/10.1016/j.compedu.2019.103798
Tikva, C., & Tambouris, E. (2021). Mapping computational thinking through programming in K-12 education: A conceptual model based on a systematic literature review. *Computers & Education*, *162*, 104083. https://doi.org/10.1016/j.compedu.2020.104083
Tucker, A. (2003). A model curriculum for K–12 computer science: Final report of the ACM K–12 task force curriculum committee. ACM. https://dl.acm.org/citation.cfm?id=2593247
Van Borkulo, S., Chytas, C., Drijvers, P., Barendsen, E., & Tolboom, J. (2021). Computational thinking in the mathematics classroom: Fostering algorithmic thinking and generalization skills using dynamic mathematics software. *Proceedings of the 16th Workshop in Primary and Secondary Computing Education*, 1–9. https://doi.org/10.1145/3481312.3481319
Weintrop, D. (2018). *Defining, Designing, and Documenting Computational Thinking Across K-12 Education*. https://repository.isls.org/bitstream/1/874/1/519.pdf. Accessed Jan 2024.
Weintrop, D., Beheshti, E., Horn, M., Orton, K., Jona, K., Trouille, L., & Wilensky, U. (2016). Defining computational thinking for Mathematics and Science classrooms. *Journal of Science Education and Technology*, *25*. https://doi.org/10.1007/s10956-015-9581-5
Wing, J. (2006). Computational thinking. *Communications of the ACM*, *49*(3), 33–35. https://doi.org/10.1145/1118178.1118215
Wing, J. (2011). Research notebook: Computational thinking—what and why. *The link Magazine*, *6*, 20–23.
Wise, A. F. (2019). Educating data scientists and data literate citizens for a new generation of data. *Journal of the Learning Sciences*, 29(1), 165–181. https://doi.org/10.1080/10508406.2019.1705678.
Witherspoon, E. B., Higashi, R. M., Schunn, C. D., Baehr, E. C., & Shoop, R. (2017). Developing computational thinking through a virtual Robotics Programming Curriculum. *ACM Transactions on Computing Education*, *18*(1), 4–20. https://doi.org/10.1145/3104982
Wolfram, C. (2020). The Math(s) fix. *An education blueprint for the AI age*. Wolfram Media, Inc.
Wong, G. K., & Jiang, S. (2018). Computational thinking education for children: Algorithmic thinking and debugging. *2018 IEEE International Conference on Teaching, Assessment, and Learning for Engineering (TALE)*, 328–334. https://doi.org/10.1109/TALE.2018.8615232

Yaşar, O. (2018). A new perspective on computational thinking. *Communications of the ACM*, *61*(7), 33–39. https://doi.org/10.1145/3214354

Zhang, S., Wong, G. K. W., Pan, G., & Pilot Testing. (2021). Computational thinking test for lower primary students: Design principles, content validation, and. *2021 IEEE International Conference on Engineering, Technology &* Education *(TALE)*, Wuhan, Hubei Province, China, 2021 (pp. 345–352). https://doi.org/10.1109/TALE52509.2021.9678852